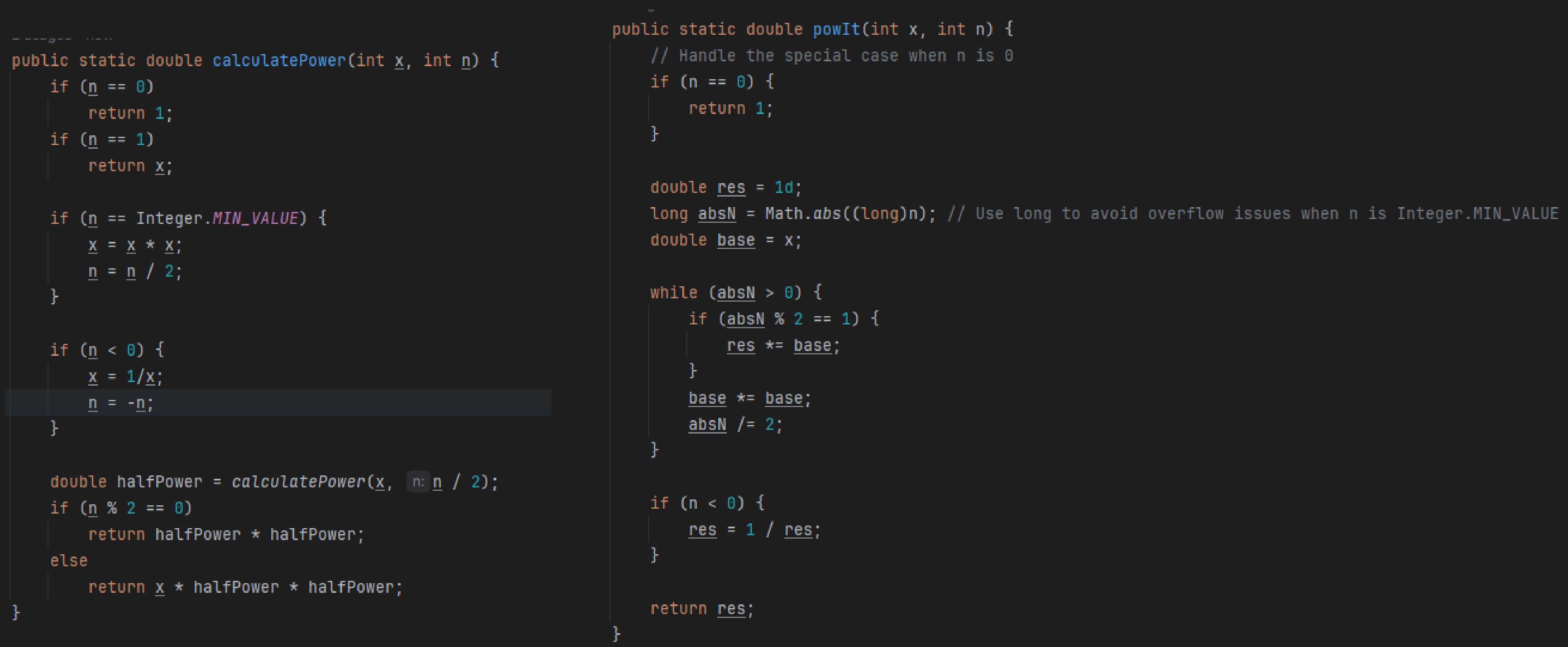

```java
public static double calculatePower(int x, int n) {
    if (n == 0)
        return 1;
    if (n == 1)
        return x;

    if (n == Integer.MIN_VALUE) {
        x = x * x;
        n = n / 2;
    }

    if (n < 0) {
        x = 1/x;
        n = -n;
    }

    double halfPower = calculatePower(x, n: n / 2);
    if (n % 2 == 0)
        return halfPower * halfPower;
    else
        return x * halfPower * halfPower;
}
```

```java
public static double powIt(int x, int n) {
    // Handle the special case when n is 0
    if (n == 0) {
        return 1;
    }

    double res = 1d;
    long absN = Math.abs((long)n); // Use long to avoid overflow issues when n is Integer.MIN_VALUE
    double base = x;

    while (absN > 0) {
        if (absN % 2 == 1) {
            res *= base;
        }
        base *= base;
        absN /= 2;
    }

    if (n < 0) {
        res = 1 / res;
    }

    return res;
}
```